AN APPROACH TO PROBABILISTIC RETRIEVAL*

C.T. Yu
Dept. of Information Engrg.
Univ. of Illinois at Chicago Circle
Chicago, IL 60680 U.S.A.

and

K.Lam
Dept. of Statistics
Hong Kong University
Hong Kong

ABSTRACT

The objective is to relate the effectiveness of retrieval, the fuzzy set concept and the processing of Boolean query. The use of a probabilistic retrieval scheme is motivated. It is found that there is a correspondence between probabilistic retrieval schmes and fuzzy sets. A fuzzy set corresponding to a potentially optimal probabilistic retrieval scheme is obtained. Then the retrieval scheme for the fuzzy set is constructed.

INTRODUCTION

The effect of term weights on the performance of queries was analyzed in [24] , where it was shown that queries whose terms having higher "precision values" are assigned heavier weights yield better retrieval results than queries whose terms are assigned the same weights, under the assumption that terms are distributed independently. Thus, the precision value of a term characterizes the usefulness of the term in retrieval. This result was supported in [13], whre it was shown that if terms of a query are assigned weights proportional to the logarithm of their precision values, then optimal retrieval results are obtained under the same term independence assumption. When terms are distributed dependently, the incorporation of the term dependence into the retrieval process yields better retrieval results [9,20,23]. Even more general condition exists for the construction of the optimal queries [5,8,21]. The above results assume that certain parameter values (e.g. those needed to compute the term precision values) are known. When these values are not known, they may be estimated by relevance feedback [5,7,15,22] where the user identifies each retrieved document as either relevant or irrelevant, and input the information to the system. Where relevance feedback can not be employed (e.g. a user submits a query the first time), various attempts have been made [6,17,18,25] to yield reasonable retrieval results. All these techniques are used when the user's queries are expressed as sets of keywords.

When a user's query is expressed as a Boolean expression, various approaches have been suggested [3,4,10,12,19]. The use of fuzzy set seems to be promising, though there was criticism on how the fuzzy set was used in information retrieval [2,14]. Very little is known how the assignment of weights to Boolean queries affects retrieval effectiveness nor how the use of fuzzy set influences retrieval effectiveness. This thesis relates the use of fuzzy set retrieval to the effectiveness of Boolean queries processing.

In Section 2, the use of a probabilistic retrieval scheme (PRS) is motivated. Our objective is to find a potentially optimal PRS. In Section 3, it is found that a PRS yields a fuzzy set and conversely. An operation is introduced such that repeated applications of the operation on a given fuzzy set yield a fuzzy set corresponding to a potentially optimal PRS. In Section 4, three retrieval schemes are compared using the results of Section 3. some concluding remarks are given in Section 5.

2. MOTIVATION

When a user submits a query, the collection of all documents, say C, is partitioned into subsets, each subset has a distinct combination of terms specified by the query. For example, if the query statement is [(t_1 OR t_2) AND (t_3 OR t_4)], where t_i represents the i-th term, then with respect to the query, the subsets are $\{(x_1,x_2,x_3,x_4) \mid x_i \in \{0,1\}$, where a '1' in the i-th position denotes the presence of the i-th term, $1 \leq i \leq 4\}$. We shall view each subset of documents as a single document because any two documents in the subset have the same combination of terms and these documents can hardly be differentiated from each other with respect to the query. From the query specification, one can usually deduce that certain documents are likely to have at least as high probabilities of relevance as some other documents. In general, a partial ordering of documents can be constructed such that D_i '$\geq$' D_j if one can deduce from the user query that the i-th document D_i has at least as high a probability of relevance as the j-th document D_j. For example, for the user query [(t_1 OR t_2) AND (t_3 OR t_4)], it is likely

---
*research supported by NSERC of Canada and N.S.F. of U.S.A.

that a document represented by (1,0,1,0) has a probability of relevance at least as high as that of another document represented by (0,0,1,1) but it is difficult to predict which of (1,0,1,0) and (0,1,0,1) has a higher probability of relevance. A partial ordering of the 16 documents with respect to the query is given in Figure 1. Note that in a partial ordering, while it is known that the probabilities of relevance of certain documents are larger than those of some other documents, the actual probabilities of relevance of the documents are not known.

Suppose the user wants to retrieve m out of a total of n documents, $n \geq m$. A choice of m documents from the partial ordering in a deterministic manner may yield supotimal retrieval results. For example, in Figure 1, if m=10, it is quite clear that the first 9 documents retrieved are $\{D_1,D_2,...,D_9\}$; however, retrieving D_{10} (D_{11}) will yield suboptimal results if D_{11} (D_{10}) has a higher probability of relevance than D_{10}(D_{11}). In fact, retrieving say $\{D_1,D_2,...,D_{10}\}$ may yield fewer expected number of relevant documents than retrieving $\{D_1,...,D_5\}$ followed by randomly choosing five out of six documents $\{D_6,D_7,...,D_{11}\}$ (each of which has the same number of terms in common with the query), each with equal chance. The former retrieval rule yields $(R_1+...+R_5+R_6+...+R_{10})$ expected number of relevant documents while the latter yields $(R_1+...+R_5+5/6+(R_6+...+R_{11}))$ where R_i is the probability of relevance of the i-th document. Thus, the latter retrieval rule yields more relevant documents if $R_1=R_2=...=R_5=1$; $R_6=R_7=...=R_9=R_{11}=0.9$; $R_{10}=0.3$. Note that this assignment of probabilities of relevance satisfies the partial ordering given in Figure 1.

Such a consideration motivates the present study on probabilistic <u>r</u>etrieval <u>s</u>chemes PRS . Each scheme assigns a <u>p</u>robability of retrieval, P_k, to each <u>subset</u>, S_k, containing m documents, $1 \leq k \leq N = c(n,m)$*. The number of relevant documents in S_k is

$$\sum_{D_i \varepsilon S_k} R_j.$$

The probability that S_k is retrieved is P_k. Thus, the expected number of relevant documents retrieved by a PRS is

$$\sum_{k=1}^{N} P_k \left(\sum_{D_j \varepsilon S_k} R_j \right).$$

This is referred to as the <u>performance</u> of the scheme. Our aim is to find a probabilistic retrieval scheme, say PRS1, such that

1) The performance of PRS1 is always better than that of the random PRS (which assigns equal probabilities of retrieval to all subsets of m documents) for any set of probabilities of relevance $\{R_i, 1 \leq i \leq n\}$ satisfying the given partial ordering. Note that a deterministic retrieval scheme may not have this property. (See Figure 2.)

2) The performance of PRS1 is better than that of any other PRS, for at least one set of probabilities of relevance satisfying the given partial ordering. In other words, no ther PRS is always better than PRS1.

Obviously, it would be nice to have a PRS which is always better than oany other PRS. However, this may not be possible. In fact, it is possible that a PRS is better than another PRS under a set of probabilities of relevance of the documents but the former PRS is worse than the latter PRS under another set of probabilities of relevance while both sets of probabilities of relevance of the documents satisfy the given partial ordering. For example, suppose two documents are to be retrieved for the partial ordering in Figure 1. A PRS which assigns non-zero probabilities of retrieval P_1,P_2, P_3 and P_4 to $\{D_1,D_2\}$, $\{D_1,D_3\}$, $\{D_1,D_4\}$, $\{D_1,D_5\}$ respectively,

$$\sum_{i=1}^{4} P_i = 1,$$

will retrieve fewer relevant documents than another PRS which assigns probabilities of retrieval $P_1+\Delta$, $P_2-\Delta$, P_3 and P_4 to the four sets, $\Delta>0$, if D_2 has a higher probability of relevance than D_3. But the latter PRS will retrieve fewer relevant documents than the former PRS if D_3 has a higher probability of relevance than D_2.

A PRS satisfying properties (1) and (2) above is a <u>potentially optimal PRS</u>. Such a PRS can be obtained as described in the next section.

3. OBTAIN A POTENTIALLY OPTIMAL PRS

Instead of assigning a probability of retrieval, P_k, to each <u>subset</u> S_k, a PRS can be viewed as assigning a probability of retrieval, r_j, to each <u>document</u> D_j, $1 \leq j \leq n$. In other words, the retrieved set is a fuzzy set 26 of n documents, wehre the "degree of membership of the j-th document in the retrieved set" denotes its probability of being retrieved. More precisely, the expected number of relevant documents retrieved by a PRS can be written as

$$\sum_{j=1}^{n} r_j R_j$$

as shown by the following lemma.

<u>Lemma 3.1</u>:

$$\sum_{k=1}^{N} P_k \left(\sum_{D_j \varepsilon S_k} R_j \right) = \sum_{j=1}^{n} r_j R_j \tag{3.1}$$

where r_j is the probability of retrieval of the j-th document.

Proof: Consider a two-dimensional matrix where the columns are the N=c(n,m) subsets of documents, the rows are the n documents and the (j,k)-th entry of the matrix =

$$\begin{cases} P_k R_j \text{ if } D_j \text{ is in the k-th subset, } S_k, \\ 0 \quad \text{otherwise.} \end{cases}$$

The left hand side of equation (3.1) is the summing of all the entries in the matrix column by column. When the entries in the j-th row are summed, we obtain =

$$\sum_{D_j \varepsilon S_k} P_k R_j = \left(\sum_{D_j \varepsilon S_k} P_k \right) R_j,$$

* c(n,m) is the number of combinations of choosing m out of n things.

where the summation is over all subsets containing the j-th document. Since r_j is the probability of retrieval of the j-th document, $r_j = \sum_{D_j \epsilon S_k} P_k$.

When j ranges from 1 to n, that is, the entries in the matrix are summed row by row, the right hand side of the euqation is obtained. □

In order to find a PRS satisfying condition (1), we initialize $r_i = m/n$, $1 \leq i \leq n$. This is the situation where each document has the same probability of being retrieved. In other words, this is a fuzzy set iwth the degree of membership of each document being the same. It implies that each subset of m documents has the same probability of being retrieved, which corresponds to the random PRS. An iterative operation is now introduced. This operation increases the probability of retrieval of a document having a higher probability of retrieval of another document having a lower probability of relevance by the same amount. As a result, the expected number of relevant documents retrieved is increased. Specifically, the operation is:

Operation A:

For any documents D_i, D_j in C if D_i '>' D_j, the probability of retrieval of the j-th document $r_j > 0$ and the probability of retrieval of the i-th document $r_i < 1$, then a $\Delta > 0$ can be chosen so that the new r_i, denoted by r_i', and the new r_j, denoted by r_j', satisfy

$$r_i' = r_i + \Delta$$

$$r_j' = r_j - \Delta \quad \text{with } 0 \leq r_i', r_j' \leq 1.$$

Figure 3(a) and 3(b) show the effect of executing the operation once for $\Delta = 1/5$ and $m = 2$.

The above operation changes a fuzzy set into another fuzzy set. We want to show that the new fuzzy set corresponds to a PRS. Specifically, given a set of r_i, $1 \leq i \leq n$, $0 \leq r_i \leq 1$,

$$\sum_{j=1}^{n} r_j = m,$$

we want to establish the existence of a set of probabilities of retrieval of the subsets, P_k, $1 \leq k \leq N$, $0 \leq P_k \leq 1$,

$$\sum_{k=1}^{N} P_k = 1$$

such that

$$\sum_{D_j \epsilon S_k} P_k = r_j \quad , 1 \leq j \leq n. \tag{3.2}$$

The above equation says that the probability of retrieval of the j-th document is the sum of the probabilities of retrieval of the subsets S_k which contain the j-th document. (3.2) can be rewritten in a matrix form, where the rows are the n documents, the columns are the N subsets and the (j,k)-th element of the matrix denotes the presence or absence of the j-th document in the k-th subset. More precisely, (3.2) is equivalent to:

$$\begin{bmatrix} a_{11} & a_{12} \cdots a_{1k} \cdots a_{1N} \\ \vdots & \vdots \\ a_{j1} & a_{j2} \cdots a_{jk} \cdots a_{jN} \\ \vdots & \vdots \\ a_{n1} & a_{n2} \cdots a_{nk} \cdots a_{nN} \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ \vdots \\ P_N \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ \vdots \\ r_n \end{bmatrix} \tag{3.3}$$

where $a_{jk} = \begin{cases} 1 & \text{if } D_j \text{ is in } S_k \\ 0 & \text{otherwise.} \end{cases}$

Clearly, (3.3) can be rewritten as (by viewing each column of the matrix as a vector

$$P_1A_1 + P_2A_2 + \ldots + P_NA_N = r \tag{3.4}$$

where $A_j = \begin{bmatrix} a_{1j} \\ \vdots \\ a_{nj} \end{bmatrix}$ and $r = \begin{bmatrix} r_1 \\ \vdots \\ r_n \end{bmatrix}$

The existence of a set of P's satisfying (3.4) and the constraints imposed on the r's and the P's is guaranteed by an application of Choquet's theorem [11].

A version of Choquet's theorem: Let X be a closed and bounded (see, for example, [1]) convex subset with finitely many extreme points on the n-dimensional space R^n, and let $x_0 \epsilon X$. Then x_0 can be written as a convex combination of the extreme points of X.

Intuitively, a convex set is one where for any two points in the set, any point in the straight line segment joining the two points is also in the set. A convex combination of a number of points is $\{x_1, \ldots, x_m\}$ is

$$\sum_{j=1}^{m} q_j x_j ,$$

where $0 \leq q_j \leq 1$ and

$$\sum_{j=1}^{m} q_j = 1.$$

An extreme point is a point which cannot be expressed as a convex combination of other points. In Figure 4, the triangle is a convex set. The extreme points are the "corners" and are labelled {1,2,3}. Any point in the triangle can be written as a convex combianation of the extreme points. For example, point 4 is a convex combination of points 1 and 2; point 5 is convex combination of 3 and 4 and is therefore a convex combination of points 1, 2 and 3.

To apply the theorem, the following lemma is established.

Lemma 3.2: Let X be $\{(s_1, \ldots s_n) | \sum_{j=1}^{n} s_j = m, 0 \leq s_j \leq 1\}$. Then (i) X is a closed and bounded convex set, (ii) each A_j is an extreme point in X, (iii) the only extreme points are the A's, $1 \leq j \leq N$ and (iv) r is in X.

Thus, (3.4) is equivalent to "r is a convex combination of the extreme points A_j, $1 \le j \le N$". Then Choquet's theorem guarantees the existence of the P's satisfying (3.4).

Operation A assigns a higher probability of retrieval to a document D_i and a lower probability of retrieval D_j when the probability of relevance of D_i is at least as high as that of D_j. Intuitively, this yields at least as many relevant documents. It is easy to show that

Lemma 3.3: The performance of the new PRS (after a successful execution of the opeation) is at least as good as that of the old PRS (before the execution of the operation).

Proof: Let r_k be the probability of retrieval of D_k in the old PRS, r_k' be the probability of retrieval of D_k in the new PRS, $1 \le k \le n$. Assume we applied the operation A on the document pair D_i, D_j where D_i '$\geq$' D_j, $r_i < 1$ and $r_j > 0$. Since $r_i < 1$, $r_j > 0$, there exists a $\Delta > 0$ such that $r_i' = r_i + \Delta$, $r_j' = r_j - \Delta$ and $0 \le r_i'$, $r_j \le 1$. The probabilities of retrieval of other documents are unchanged.

The performance of the new PRS = $\sum_{t=1}^{n} r_t' R_t$

$$= \sum_{t \neq i,j} r_t' R_t + (r_j - \Delta) R_j + (r_i + \Delta) R_i$$

$$= \sum_{t=1}^{n} r_t R_t + \Delta(R_i - R_j) \geq \sum_{t=1}^{n} r_t R_t = \text{the performance}$$

of the olds PRS. ∎

Since we start off with the random PRS and each execution of the operation yields a better PRS, we must end up with a better PRS. When the operation can no longer apply (i.e. for every D_i '$\geq$' D_j, either $r_i = 1$ or $r_j = 0$), a PRS is reached, say PRS2, which satisfies the following property:

CONDITION(*)

For any pair of documents, D_i and D_j in set C, D_i '$\geq$' D_j and $r_j \neq 0$ imply $r_i = 1$.

This PRS2 can be shown to possess the property that

Lemma 3.4: No other PRS is always better than PRS2, i.e. condition (2) is satisfied.

Proof: Assume PRS2 assigns probability of retrieval r_i for each document D_i and some other PRS, say PRS', assigns probability of retrieval t_i for document D_i, $1 \le i \le n$. Let $A = \{D_i \mid t_i < r_i\}$ and $A' = A \cup A_1$ where $A_1 = \{D_i \mid D_i$ '$\geq$' D_j for some D_j in A but $D_i \notin A\}$. By definition of A_1 and PRS2 satisfies Condition (*), each D_i in A_1 satisfies $r_i = 1$. Let m be the number of documents retrieved. We want to show that there always exists a set of $\{R_i \mid 1 \le i \le n\}$ satisfying the given partial ordering and the performance of PRS2 is better than that of PRS' under the set.

Case(1) $m \le |A'|$,

$$\text{Let } R_i = \begin{cases} m/|A'| & \text{if } D_i \text{ is in A'} \\ 0 & \text{otherwise.} \end{cases}$$

We now show that $\{R_i \mid 1 \le i \le n\}$ satisfies the partial ordering by establishing that for each document D_j in A', if D_i '$\geq$' D_j, then D_i is also in set A'. If D_j is in A, then since $r_j > 0$ and D_i and D_j satisfy Condition(*), $r_i = 1$. By definitions of A and A_1, D_i is in A'. If D_j is in A_1, then there is a D_k in A such that D_j '$\geq$' D_k. Since D_i '$\geq$' D_j and '$\geq$' is transitive, D_i '$\geq$' D_k. Thus D_i is in A'. We now show that PRS2 has a better performance than PRS'.

$$\sum_{i=1}^{n} r_i R_i = \sum_{i \in A_1} r_i R_i + \sum_{i \in A} r_i R_i$$
$$= \sum_{i \in A_1} 1 * R_i + \sum_{i \in A} r_i R_i$$
$$> \sum_{i \in A_1} R_i + \sum_{i \in A} t_i R_i$$
$$\geq \sum_{i \in A_1} t_i R_i + \sum_{i \in A} t_i R_i$$
$$= \sum_{i \in A'} t_i R_i \quad .$$

Case(2) $m > |A'|$,

$$\text{Let } R_i = \begin{cases} 1 & \text{if } D_i \text{ is in A'} \\ (m-|A'|)/(n-|A'|) & \text{otherwise.} \end{cases}$$

Then $R_i \mid 1 \le i \le n$ satisfies the partial ordering as in Case(1).

$$\sum_{i=1}^{n} r_i R_i = \sum_{i \in A'} r_i + \sum_{i \notin A'} r_i((m-|A'|)/(n-|A'|))$$
$$= \sum_{i \in A'} r_i + ((m-|A'|)/(n-|A'|)) \sum_{i \notin A'} r_i$$

$$\text{Let } \Delta = \sum_{i \in A'} r_i - \sum_{i \in A'} t_i > 0,$$

then since

$$\sum_{i \in A'} r_i + \sum_{i \notin A'} r_i = m = \sum_{i \in A'} t_i + \sum_{i \notin A'} t_i,$$
$$\sum_{i \notin A'} t_i - \sum_{i \notin A'} r_i = \Delta.$$

Thus the above expression

$$= (\Delta + \sum_{i \in A'} t_i) + ((m-|A'|)/(n-|A'|))(\sum_{i \notin A'} t_i - \Delta)$$
$$= \Delta\,(1-((m-|A'|)/(n-|A'|)) + \sum_{i=1}^{n} t_i R_i$$
$$> \sum_{i=1}^{n} t_i R_i, \text{ for } n > m. \quad ∎$$

Thus, repeated applications of the operation, until the operation can no longer applies, yields a PRS satisfying both Conditions (1) and (2). That is, a potentially optimal PRS is obtained.

4. SCHEMES COMPARISON

In this section, we present three natural retrieval schemes. Under a certain partial ordering of documents induced by a given Boolean query, the performances of three schemes are compared.

4.1 Query Representation

A Boolean query Q can always be expressed as a number of OR-groups, connected together by AND operators, each group being a Boolean expression of indexed terms connected by OR operators. Specifically, the query Q can be written as follows:

$$Q = \mathop{\mathrm{AND}}_{p=1}^{M}(O_p) = \mathop{\mathrm{AND}}_{p=1}^{M}(\mathop{\mathrm{OR}}_{q=1}^{M_p} t_{pq})$$

where M is the number of OR-groups in Q, O_p is the p-th OR-group, m_p is the number of terms in O_p, and t_{pq} is the q-th term in the p-th OR-group O_p.

Example 4.1:

If Q = (t_1 OR t_2) AND (t_3 OR t_4) AND (t_1 OR t_5), then there are 3 OR-group O_1 = (t_1 OR t_2), O_2 = (t_3 OR t_4) and O_3 = (t_1 OR t_5).

The set of OR-groups in common between Q and D_i, namely G_i, is the set of OR-groups of Q, each of which has a term occurring in D_i. The number of terms in common between Q and D_i, namely $|T_i|$, is the number of occurrences of the terms of Q, each of which occurs in D_i.

Example 4.2:

From Example 5.1, let $D_1 = \{t_1, t_2, t_6\}$.
$|T_1| = 3$ since t_1 occurs in O_1 and O_3, and t_2 occurs in O_1.
$G_1 = \{O_1, O_3\}$.

Now we consider the following natural retrieval schemes RS1, RS2, and RS3:

RS1: Retrieve documents in descending order of the number of terms in common with the query Q. Documents with the same number of terms in common with the query are retrieved with equal probabilities.

RS2: Retrieve documents in descending order of the number of terms in common with the query Q. Documents with the same number of terms in common iwth the query are then retrieved in descending order of the number of OR-groups in common with Q. Documents with the same number of terms and same number of OR-groups in common with Q are retrieved with equal probabilities.

RS3: Retrieve documents in descending order of the number of OR-groups in common with the query Q. Documents with the same number of OR-groups in common iwth Q are then retrieved in descending order of the number of terms in common with Q. Documents with the same number of terms and same number of OR-groups in common iwth Q are retrieved with equal probabilities.

Assume a user issues a query Q. A partial ordered relation for pairs of documents in C can be deduced from Q based on the following intuition. It is likely that a document D_i which has more OR-groups in common iwth Q than another document D_j is more likely to be relevant to the user submitting query Q than D_j. Specifically, the partial ordered relation is given by:

A document D_i which has more number of OR-groups in common iwth the query Q than another document D_j has at least as high a probability of relevance as D_j. That is, D_i '$\geq$' D_j if and only if $|G_i| > |G_j|$.

Example 4.3:

From Example 5.2, let D_2 = t_1, t_3 .
We have D_2 '$\geq$' D_1 since $|G_2| = 3 > 2 = |G_1|$.

Under the above partial ordering it can be shown that RS3 performs better than RS2, which is better than RS1.

CONCLUSION

The use of a probabilistic retrieval scheme (PRS) is motivated. It is applied to the processing of Boolean queries. Our aim is to obtain a potentially optimal PRS. To achieve this, a correspondence between PRS and fuzzy sets is established. A process to obtain a fuzzy set corresponding to a potentially optimal PRS is presented. Then, a potentially optimal PRS is constructed from the fuzzy set.

Finally, the performances of some natural retrieval schemes are compared using a partial ordering deduced from a given Boolean query.

The main contributionsof the work presented here are
(1) a relationship between a retrieval scheme and its retrieval effectiveness is established analytically;
(2) the use of fuzzy set, which has been employed by earlier researchers but not related to the effectiveness of retrieval, fits into the development of (1) naturally; and
(3) a conceptually very simple process to obtain a potentially optimal PRS is provided. This procedure is independent of the given partial ordering. Thus, if a better partial ordering (than the one given here) is obtained by another interpretation of a Boolean query or by re-evance feedback, the procedure given here can still be applied.

REFERENCES

[1] Apostol, T.M., "Mathematical Analysis", Addison-Weslay 1957.

2 Bookstein, A. "Perils of Merging Boolean and Weighted Retrieval System" JASIS 29 (May 1978) 156-157.

3 Bookstein, A. "Fuzzy Request: An Approach to Weighted Boolean Searches", JASIS (June 1980).

4 Buell, D. and Kraft, D.H., "A Model for a Weighted Retrieval System", JASIS (to appear).

5 Chow, D., Yu, C.T., "On the Construction of Feedback Queries", JACM (to appear).

6 Croft, W.B., Happer, D.J., "Using Probabilistic Models of Document Retrieval Without Revelance Information", Journal of Documentation 35, (DEC. 1979) 285-289.

7 Harper, D.J., Van Rijsbergen, C.J., "An Evaluation of Feedback in Document Retrieval Using Co-occurence Data", Journal of Documentation 34, 1978, 189-216.

8 Kraft, D., Bookstein, A., "Evaluation of Information Retrieval Systems: A Decision Theory Approach", JASIS 1978, 31-40.

9 Lam, K., Yu, C.T., "A Cluster Search Algorithm Incorporating Arbitrary Term Dependencies", Department of Information Engineering, University of Illinois at Chicago Circle, 1978.

10 Noreault, T., Koll, M., McGill, M.J., "Automatic Ranked Output from Boolean Searches in SIRE", JASIS (Nov. 1977), 333-339.

11 Phelps, Robert, "Lecture on Choquet's Theorem", D. Van Nostrand Computer, Inc., 1966.

12 Radecki, T., "Mathematical Model of Information Retrieval Systems Based on a Concept of Fuzzy Thesaurus", Information Processing and Management (Sept. 1976) 313-318.

13 Robertson, S.E.; Sparck Jones, K., "Relevance Weighting of Search Terms" JASIS (May-June 1976) 129-146.

14 Robertson, S.E. "On the Nature of Fuzz: A Diatribe" JASIS (Nov. 1978) 304-307.

15 Rocchio, J.J. "Relevance Feedback in Information Retrieval" in Salton, G. (ed.) the SMART Retrieval System. New Jersey, Prentice -Hall, 1971.

16 Salton, G. "Automatic Information Organization and Retrieval" New York, McGraw-Hill 1968.

17 Salton, G.; Yang, C.S.; Yu, C.T. "A Theory of Term Importance in Automatic Text Analysis" JASIS, (Jan. 1975) 33-44. (Also reprinted in "Key Papers in Information Science", ASIS, 1980, 293-304.)

18 Spark Jones, K. "Search Term Relevance Weighting Given Little Relevance Information" J. of Documentation, 35, 1979, 30-48.

19. Tahani, V. "A Fuzzy Model of Document Retrieval System" Information Processing and Management (May 1976) 177-188.

20 Van Rijsbergen, C.J. "A Theoretical Basis for the Use of Co-occurrence Data in Information Retrieval" J. of Documentation (June 1977) 106-119.

21 Van Rijsbergen, C.J. "Information Retrieval" Butterworth, 2nd edition, 1979.

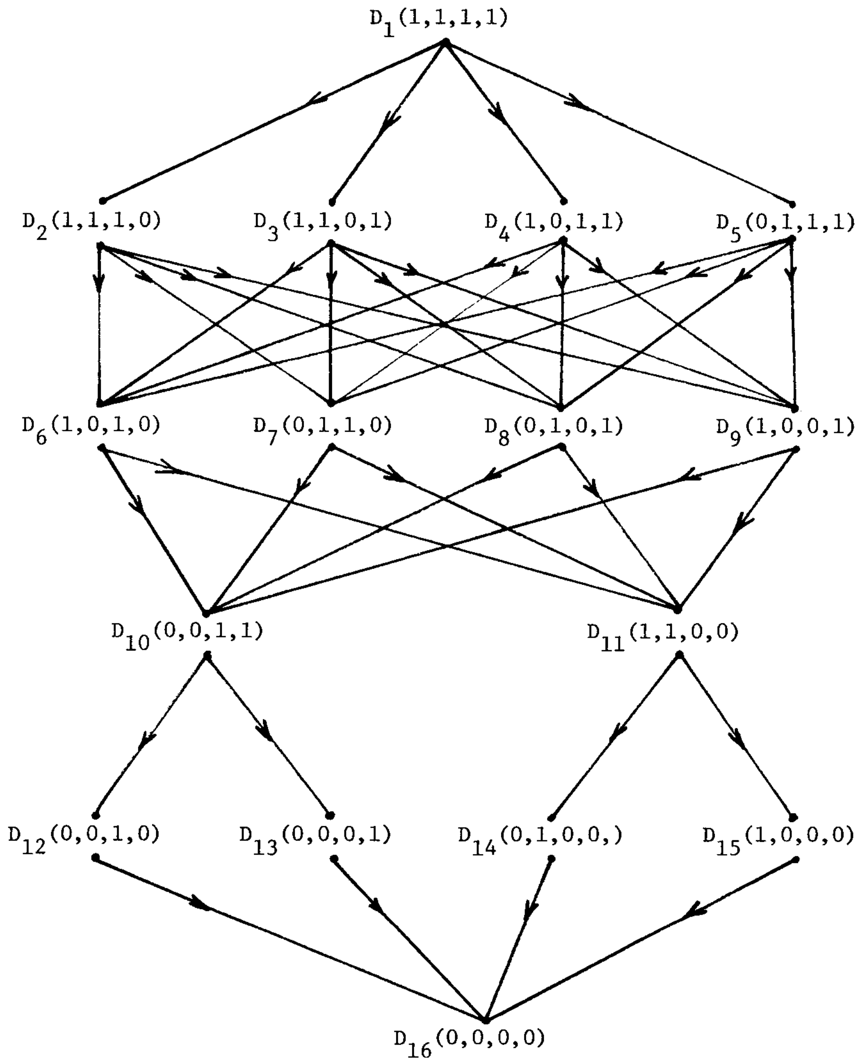
22 Yu, C.T.; Luk, W.S.; Cheung, T.Y. "A Statistical Models for Relevance Feedback in Information Retrieval: JACM (April 1976) 273-286.

23 Yu, C.T.; Luk, W.S.; Siu, M.K "On Models of Information Retrieval Processes" Information System, Vol 4,3, 1979, 205-218.

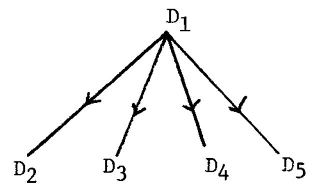
24. Yu, C.T.; Salton, G., "Precision Weighting: an Effective Automatic Indexing Method" JACM (Jan. 1976) 76-88.

25 Yu, C.T.; Lam, K.; Salton, G. "Term Weighting in Information Retrieval Using the Term Precision Model" JACM (to appear).

26 Zadek, L.A., "Fuzzy Set", Information and Control (June 1965) 336-353.



<u>Figure 1</u>: A partial ordering of the documents with respect to the query (t_1 or t_2) and (t_3 or t_4). $D_1 \mathbin{`\geq'} D_2$; $D_6 \mathbin{`\geq'} D_{10}$ etc.



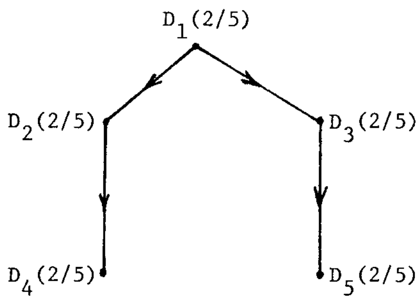
3 documents are to be retrieved.

A deterministic retrieval scheme might retrieve D_1, D_2, D_3 .

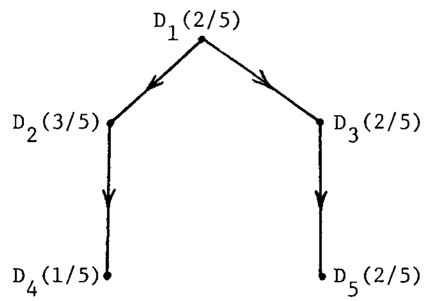
Suppose R_1=1, R_2= 0, R_3=o, R_4=1, R_5=1.

Then the deterministic scheme retrieves 1 relevant document only, while a random retrieval scheme retrieves 3/5(1+0+0+1+1) =9/5 relevant documents.

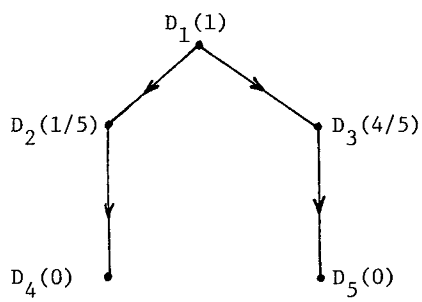
Figure 2 : Illustrates that a deterministic retrieval scheme may not always be better than a random retrieval scheme.



(a) Before applying the operation, the PRS is the random PRS. The numbers in paranthesis are the probabilities of retrieval.



(b) A PRS obtained after one execution of the operation on the random PRS with $\Delta=1/5$. The two documents acted on by the operation are D_2 and D_4.



(c) A possible PRS after repeated applications of the operation. This PRS satisfies conditions (i) and (ii).

Figure 3: Illustrates the operation on a partial ordering. The number of retrieved documents, i, is 2.

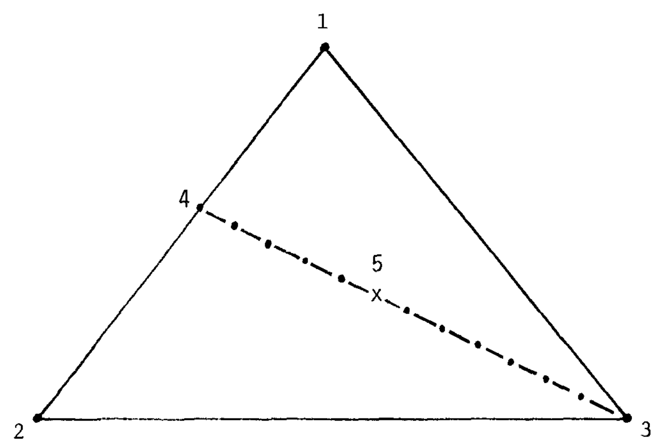


Figure 4: The extreme points of the triangle are {1,2,3}. Any point in the triangle is a convex combination of {1,2,3}.